

Netherlands
organization for
applied scientific
research

FEL

TNO Physics and Electronics
Laboratory

①

report no.
FEL-90-A080

copy no.

title

User Manual of the Force Structure Model Unit Type
Editor TDM

AD-A229 697

author(s):

Vdg. Drs. K. Kiel

DTIC
ELECTE
DEC 18 1990
S B D

date : November 1990

classification

title : unclassified

abstract : unclassified

reporttext : unclassified

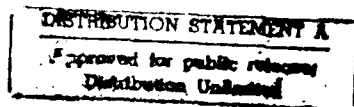
appendices A and B : unclassified

no. of copies : 25

no. of pages : 26 (incl. appendices, excl. RDP + distribution list)

appendices : 2

All information which is classified according to Dutch
regulations shall be treated by the recipient in the same
way as classified information of corresponding value in
his own country. No part of this information will be
disclosed to any party.



90 12 17 093

report no. : FEL-90-A080
title : User Manual of the Force Structure Model Type Definition Module TDM

author(s) : Vdg. Drs. R. Kiel
institute : TNO Physics and Electronics Laboratory

date : November 1990
NDRO no. : A89KL619
no. in pow '90 : 701.1

Research supervised by : Drs. P.A.B. van Schagen
Research carried out by : Vdg. Drs. R. Kiel
=====

ABSTRACT (UNCLASSIFIED)

The Type Definition Module TDM is part of the combat simulation model Force Structure Model FSM and has been developed under contract A89KL619. The purpose of TDM is to provide a way of interactively creating and modifying so-called UNIT_TYPES. These UNIT_TYPES are used as templates for creating the UNITS for a FSM simulation run. The UNIT_TYPES are stored in two Type Definition Files: the systemwide available PRE_DEFINED_TYPES and the locally available USER_DEFINED_TYPES. The report serves as the user manual for TDM.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist.	Avail and/or Special
A-1	



rapport no. : FEL-90-A080
titel : Gebruikers Handleiding van de Type Definition Module TDM van het
Gevechts Simulatie
Model FSM
auteur(s) : Vdg. Drs. R. Kiel
instituut : Fysisch en Elektronisch Laboratorium TNO
datum : november 1990
hdo-opdr.no. : A89KL619
no. in iwp : 90 : 701.1
Onderzoek uitgevoerd o.l.v : Drs. P.A.B. van Schagen
Onderzoek uitgevoerd door : Vdg. Drs. R. Kiel

=====

SAMENVATTING (ONGERUBRICEERD)

De Type Definition Module TDM is een onderdeel van het gevechtssimulatiemodel Force Structure Model FSM en is ontwikkeld onder contract A89KL619. Het doel van TDM is het interactief kunnen creëren en onderhouden van Eenheden Types (UNIT_TYPES). Deze eenheden types vormen de onderliggende structuur waarop de eenheden (UNITS) voor een FSM simulatie gebaseerd worden. De eenheden types worden in twee Type Definition Files opgeslagen. De eerste is de algemeen toegankelijke PRE_DEFINED_TYPES file, de andere de bij de gebruiker gedefinieerde USER_DEFINED_TYPES file. Dit rapport fungeert als de gebruikershandleiding voor TDM.

ABSTRACT	2
SAMENVATTING	3
CONTENTS	4
1 INTRODUCTION	5
2 UNIT_TYPES	6
2.1 Introduction	6
2.2 The structure of UNIT_TYPES	7
2.3 Type Definition Files	9
2.4 TDM in FSM	9
3 TDM USER INTERFACE	12
3.1 The VDU screen	12
3.2 The keyboard	14
4 TDM USER MANUAL	16
4.1 Before starting...	16
4.2 Starting...	17
4.3 TDM levels	17
4.4 The FILE level	18
4.5 The TYPE level	19
4.6 The WEAPON level	20
4.7 The AMMO level	21
4.8 The ASSIGNMENTS level	22
REFERENCES	24
APPENDIX A EXAMPLE PARAMETER FILES	
APPENDIX B TDM ERRORS	

1 INTRODUCTION

The Type Definition Module TDM is part of the combat simulation model Force Structure Model FSM and has been developed under contract A89KL619. The purpose of TDM is to provide a way of interactively creating and modifying so-called UNIT_TYPES. These UNIT_TYPES are used as templates for creating the UNITS for a FSM simulation run. The UNIT_TYPES are stored in two Type Definition Files: the system-wide available PRE_DEFINED_TYPES and the locally available USER_DEFINED_TYPES.

This report describes the structure of UNIT_TYPES and the way TDM manipulates this structure. Also the relation of UNIT_TYPES to the rest of FSM is briefly described. For a detailed description of all parts of FSM and the concepts of the underlying Wargame Construction Set the reader is referred to [Schagen 1] and [Joppe 1].

Chapter 2 describes UNIT_TYPES and the Type Definition Files in which they are stored. The relation of TDM with such programs as the Unit Definition Program (UDP) and FSM itself is also described here. In chapter 3 some general information on the "User Interface", i.e. on how to use the VDU (the Video Display Unit, or screen) and keyboard is given. Finally, chapter 4 explains how to use TDM in detail.

The Type Definition Module (TDM) is written in the Ada programming language, and runs on VAX computers with the VMS operating system.

2 UNIT_TYPES

This chapter describes in detail the units that participate in the combat simulation, the structure of the UNIT_TYPES and the Type Definition Files in which the UNIT_TYPES are stored. Also the relation of TDM with the Unit Declaration Program (UDP) and the rest of FSM is described.

2.1 Introduction

One of the most important objects in a combat simulation model is a UNIT. Units are the entities that participate in simulation runs and that interact with each other (e.g. engage in combat). UNIT_TYPES are the templates that are used to create these units.

In FSM units are modelled as a collection of data and processes. Units 'know' things about themselves and other (possibly enemy) units. Units can also 'talk' to each other by sending messages over networks that exist between them.

There are two main kinds of units: commanding and non-commanding units. Commanding units represent the brain of the military commander, while non-commanding units represent the physical resources (like tanks, guns, etc.).

In FSM (or better: in the underlying Wargame Construction Set WCS) units are represented by sets of data items called attributes. Each attribute holds a bit of information about the unit (or another unit). Typical attributes are: color, position, level, etc. During combat simulation the attributes of a unit are continuously being consulted and updated by the model. It will be clear that the user can only update the value of attributes belonging to a unit. Adding an attribute to, or removing an attribute from a unit means creating a whole new kind of unit. The assignment of attributes to units is therefore performed by the builder of a specific model. The model builder can define sets of attributes called UNIT GROUPS. Each unit group has its own set of attributes. The unit group also determines the actions that can be carried out by the units of that group. The following unit groups are currently defined: MANOEUVRE, ARTILLERY and their commanding counterparts C_MANOEUVRE and C_ARTILLERY.

As has been stated above, the UNIT_TYPES are the templates that can be used to create units of a certain kind. A UNIT_TYPE determines to which unit group the unit will belong and can give default values for some of its attributes.

The UNIT_TYPE also defines the physical resource that units of that type will have. These physical resources consist of a number of carriers, one or more weapons and for each weapon one or more kinds of ammunition.

2.2 The structure of UNIT_TYPES

Before the Type Definition Module (TDM) can be used, the user should be familiar with the general structure of a UNIT_TYPE. This structure is shown in figure 2.1.

```

UNIT_TYPE =
  NAME      : unit_type_names (KEY)
  DEFLEV    : definition_levels
  GROUP     : unit_groups
  CARRIER =
    NAME     : names
    AMOUNT   : natural
  WEAPONS =
    NAME     : names
    AMOUNT   : natural
    MAX_AMMO_AMOUNT : natural
    MAX_AMMO_STOCK  : natural
    AMMO = NAME  : names
           AMOUNT : natural
           STOCK  : natural
  ASSIGNMENTS = ATTRIBUTE : attribute_names (KEY)
                   VALUE   : wcs_types

```

Fig. 2.1: UNIT_TYPE structure

In this structure the following fields are defined:

NAME	This field uniquely identifies a UNIT_TYPE and is used for referring to it when creating units of a certain type for use in FSM simulation runs.
DEFLEV	This field (short for definition level) is used internally to determine the origin of a certain UNIT_TYPE. It takes one of the values PRE_DEF and USER_DEF.
GROUP	This field indicates the unit group to which a UNIT_TYPE belongs.
CARRIER	Together with its subfields this contains the information on the weapons and ammunition that are standard for a UNIT_TYPE.
ASSIGNMENTS	This field contains a list of 'ATTRIBUTE = VALUE' pairs. Whenever a new unit of a certain UNIT_TYPE is created, its assignments are effectuated.

The CARRIER field is composed of the following subfields:

NAME	This is the name of the carrier. This name also links the carrier (and thus the unit) to a certain target class in the Database Weapon Indicators (see reference [DBWI]).
AMOUNT	Defines the number of these carriers in a unit of the current type.
WEAPON NAME	The name of a weapon that is present on a carrier. This name should also be defined in the DBWI since the simulation software needs specific weapon information when units engage in combat.
WEAPON AMOUNT	The total number of weapons of this kind that is present on all carriers in the unit.
WEAPON MAX AMOUNT	The total space that is directly available for all kinds of ammunition that can be used with this weapon.
WEAPON MAX STOCK	The total space that is indirectly available for all ammunition for this weapon (used for refill).
WEAPON AMMO NAME	The name of a kind of ammunition for this weapon. Like the weapon name, the ammo name should also be defined in the DBWI.
WEAPON AMMO AMOUNT	The amount of this kind of ammunition that is directly available
WEAPON AMMO STOCK	The amount that is indirectly available.

The weapons and ammunition of a unit is structured as follows. For every kind of ammunition, there is a certain amount directly available for reloading and firing. This is the WEAPON AMMO AMOUNT. This can be thought of as a rack close to the gun. Apart from this amount, the carrier can have more storage space for that ammunition. This extra space is called the WEAPON AMMO STOCK. The unit can decide to move ammunition from the stock to the rack. This process is called refill.

For one weapon there can be more than one kind of ammunition. Each ammo kind has its own amount and stock. Because the total available space is of course limited, for each weapon a total amount (WEAPON MAX AMOUNT) and total stock (WEAPON MAX STOCK) have been defined. The WEAPON MAX AMOUNT determines the size of the rack. The sum of all ammo amounts can never exceed the max amount. Max stock has been defined similarly.

2.3 Type Definition Files

The UNIT_TYPES as described above are stored in so-called Type Definition Files (TDFs). There are two of these files. The first is the PRE_DEFINED_TYPES file. Normally there will only be one PRE_DEFINED_TYPES file, which should then be available to all users. This file contains all UNIT_TYPES that come with a certain version of the simulation software. The second TDF is the USER_DEFINED_TYPES file. Every user can have one (or even more) of these, although it is not mandatory. In this file, very specific types, or modifications of predefined types can be defined. These files are local and therefore not available to other users.

2.4 TDM in FSM

The complete Force Structure combat simulation Model (FSM) consists of a number of programs that work together sharing information about the simulation that is stored in a number of datafiles. This paragraph gives a short overview of the complete FSM system. For a detailed description the reader is referred to [Schagen 1].

Figure 2.2 shows the main components of the FSM system. The rectangles indicate programs; the circles the datafiles.

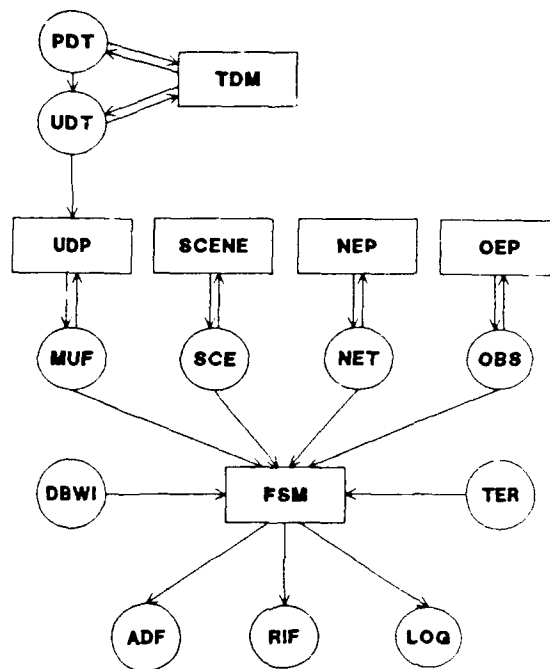


Fig. 2.2: Overview of the FSM system

Below a short description of the items occurring in figure 2.2 is given.

TDM	The TYPE DEFINITION MODULE for maintaining the Type Definition Files.
PDT	The PRE_DEFINED_TYPES file. This file contains all types that are defined on a 'system' level.
UDT	The USER_DEFINED_TYPES file. This file contains the types that are defined by a user, and are available only locally.
UDP	The UNIT DECLARATION PROGRAM. This program is used to create all the units that will be used for a certain series of simulation runs. To create units, UDP uses the information from the type definition files.
MUF	The META UNIT FILE. In this file all units, created by the UDP are stored.

SCENE	The SCENARIO EDITOR. This program is used to create a scenario. A scenario is a set of rules and guidelines determining certain aspects of the simulation. For a detailed description of scenarios and SCENE, the reader is referred to [Schagen 2].
SCE	The SCENARIO FILE, in which the complete scenario is stored.
NEP	The NETWORK ENTRY PROGRAM. This program is used to set up all connections between the units in a simulation. There has to be some kind of connection between two units to enable them to exchange messages and orders.
NET	The CONNECTIONS FILE containing the network description made with NEP.
OEP	The OBSTACLE ENTRY PROGRAM. With this program (dynamic) obstacles like mine fields and bridges can be defined.
OBS	The OBSTACLE FILE in which obstacle information is stored.
DBWI	The DATABASE WEAPON INDICATORS. This database, which is also used outside the FSM environment, contains information on carriers, weapons, ammunition and duels (see reference [DBWI]).
TER	This set of files form the TERRAIN on which the simulations take place.
FSM	The FORCE STRUCTURE MODEL. This is the core of the system, the simulation program itself. It uses all of the files that are described above as input.
ADF	The Analysis Data File generated by FSM. These files contain analysis information about simulation runs. Mainly information about duels (ATTACKER vs. TARGET) is stored. Several analysis programs are available to extract more specific information from these files.
RIF	The RESTART INFORMATION FILE. In this file all relevant information necessary for a system restart is stored. This file is also used in a MOVIE program to provide a visual playback of a simulation run.
LOG	The LOGGING file can be used for debugging purposes. For many parameters the logging can be turned on or off, depending on what programs or parts of programs are being checked.

Of course there is more to the FSM system than has been described above. Check the list of references for more information on specific items.

3 TDM USER INTERFACE

This chapter describes the way the Type Definition Module (TDM) uses the VDU (Video Display Unit) screen and the keyboard. It explains how and where information regarding the element that are currently being edited is displayed and how this information can be modified. It also explains the meaning of some of the 'hot-keys' on the keyboard.

The user interface of the Type Definition Module TDM is based on the package ATD. This 'Advanced Terminal Driver' is a set of run-time functions that provide application programs like TDM with a standard method of performing input and output on (character cell) terminals. A detailed description of ATD can be found in reference [Bakuwel 1].

3.1 The VDU screen

The layout of the TDM screen is such that all information regarding the items that are currently being edited is presented in a uniform way. An example screen is shown in figure 3.1.

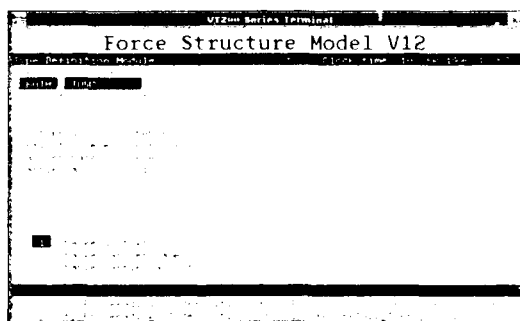


Fig. 3.1: TDM screen layout

The layout of the screen is as follows:

- Lines 1 and 2 show the current version number of the complete Force Structure combat simulation Model
- Line 3 shows the name of the program (i.e. Type Definition Module) and the date and time the program was started.

- Lines 4 through 6 form the so-called 'STATUS LINE'. This status line shows what is currently being edited and on what level this editing is taking place. In the example the Unit Type TOW2 of the Pre-defined types is being edited (note though that both files were read in). The current level is the 'type' level. Therefore all actions like SELECT, INSERT and REMOVE will operate on 'types'.
- The part of the screen between lines 8 and 21 is the general output window. This window is used for several things. First, information on the currently chosen item is displayed. This information is always displayed in the upper half of the output window. The example shows that the unit type TOW2 is of the MANOEUVRE unit group and has ONE carrier (with name PRATV).
In the lower half of the window a menu is displayed, providing access to this information.
- The last two lines of the screen show the last two messages that have been written to the logging terminal. In the example, the program has logged that it has just read both Type Definition Files.

There are several ways in which information can be feeded to TDM. These are described below. Which way will be used for a particular piece of information depends on the kind of information that is being edited.

MENUS

Menus are used to select what piece of information has to be changed. A menu consists of a list of items, consecutively numbered. An item can be selected by either using the UP and DOWN arrow keys or by directly entering the item number. A choice is confirmed (and executed) after RETURN is pressed.

SELECTORS

Selectors allow the user to select a value from a limited list of alternatives. The selector is displayed in a special selection window, which pops up on the right side of the general output window. At any time a maximum of ten alternatives is displayed. The user can scroll through the alternatives using the UP and DOWN arrow keys (one item at the time) or the NEXT and PREV SCREEN keys (one 'page' at the time). A selection can be made by pressing the RETURN key.

In several selectors a special item is defined. It is labeled '-input-' and appears as the first item in the list. If this item is selected, an input field window will pop up, allowing the user to enter a value outside the list of alternatives.

INPUT FIELDS

An input field is used when the new value of an item should be entered directly from the keyboard. In the middle of the general output window a small (one line) window pops up, containing the prompt (like 'Carrier amount?'). Underneath this prompt, another empty window pops up. In this window the current value of the item is displayed. This value can then be edited and the new value can be confirmed by pressing the RETURN key.

3.2 The keyboard

In the previous section the way TDM makes use of the VDU screen and different ways to enter input were described. Already some of the keys on the keyboard were mentioned. This section describes the meaning of these so-called 'hot keys'. In some of the descriptions the word 'level' is used. The concept of TDM levels is explained in paragraph 4.3.

HELP

Pressing this key pops up a window in which the meaning of the currently active keys is described. This key is always active except when entering input values.

RETURN

This key is used to confirm selections and input values.

ESCAPE (F20 on VT220 keyboard)

With this key any question or selection can be aborted without processing the (perhaps partial) answer that has been given so far. This key is always active.

LEFT ARROW

This key is used to move UP one level (e.g. from WEAPONS back to TYPE level). On the highest level this means leaving TDM and the user will be prompted to confirm this.

RIGHT ARROW

This key is used to move DOWN one level. E.g. after a FILE has been selected, RIGHT arrow has to be pressed to start editing TYPES.

UP / DOWN ARROW

These keys are used in menu's and selectors to point at the item that has to be selected. The current choice is highlighted.

NEXT / PREV SCREEN

These keys are used in selectors and are used to go through the list of items one page at the time (8 items in stead of 1).

SELECT

Pressing this key can be used to select an item on the current level. If for instance the user is working on the WEAPONS of a certain TYPE, pressing SELECT will show a list of currently defined weapons for that type and one of those weapons can be selected. So selection always acts on information that is already defined.

INSERT

This key can be used to insert a new item on the currently active level. In most cases TDM will present a list of items that can be inserted. The special '-input-' item makes it possible to insert something that is not available in the list. If, in the previous example, INSERT was pressed, a list would come up with all possible weapons for the unit type, together with the special '-input-' item.

REMOVE

Finally, this key gives the user the option to delete the item that has been chosen on the current level. Removing an item must always be confirmed.

The next chapter will describe how to use the Type Definition Module in detail.

4 TDM USER MANUAL

In this chapter a detailed description of how to use the Type Definition Module is given. It starts with a list of definitions that have to be made before starting TDM. Then it describes all the actions that can be performed on all levels of TDM.

4.1 Before starting...

Before the Type Definition Module is started, a number of definitions have to be made, both in the FSM_USER_FILES file as well as in the FSM_USER_PARAMETERS file. A complete description of these definition files can be found in the chapter on FSM_SYSTEM_PARS in reference [Joppe 1].

The following file/terminal logicals should be defined in FSM_USER_FILES:

FSM_USER_PARAMETERS

This file logical name points to the file containing the user defined parameters.

EML_FILE

This file will contain all error log messages and warnings. Any file name (without an extension) is valid. From the name that is specified, TDM will construct the following name:
<NAME>_<FSMVERSION>.<RUN_ID>_TDM.

PDT_FILE

This logical indicates the file containing the PRE_DEFINED_TYPES. The name should not contain an extension. Usually it will be: FSM\$DAT:[000000]PRE_DEFINED_TYPES.

UDT_FILE

This logical indicates the USER_DEFINED_TYPES file, and can be given any name (without extension).

LOG_TERM

Use this logical to indicate on which terminal the logfile has to be printed. Specifying NL: as LOG_TERM will prohibit interactive error logging. Do not specify TT: as LOG_TERM, as ATD cannot handle this correctly.

The PDT_FILE would in fact be defined in the (global) FSM_SYSTEM_FILES file, but is shown here to be complete.

Apart from these files and terminals, a number of parameters have to be defined in the FSM_USER_PARAMETERS file:

RUN_MODE

This parameter determines the mode in which the program is run, and should have the value INTERACTIVE.

RUN_ID

With this parameter specifies the run identification number. This RUN_ID is used as part of the name of the error message log file (EML_FILE).

In appendix A example FSM_USER_FILES and FSM_USER_PARAMETERS files are shown.

4.2 Starting...

After the proper logicals for files (terminals) and parameters have been defined, the Type Definition Module can be started by typing (at the VAX/VMS prompt):

```
$ RUN FSM$EXE:[version]TDM
```

or (if defined):

```
$ TDM
```

After a (short) while, the VDU screen will look as described in chapter 3. The current level is now the FILE level, and the user is ready to start editing UNIT_TYPES as will be described shortly.

4.3 TDM Levels

When using the Type Definition Module the user is always working on a certain LEVEL. The level determines what kind of information can be selected, modified, removed or inserted. The levels are hierarchically ordered. Moving from one level to another can only be done according to the hierarchical

structure (using the LEFT and RIGHT ARROW keys as described in section 3.2). The level structure is shown in figure 4.1.

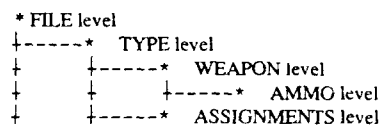


Fig. 4.1: TDM levels

The rest of this user manual is divided in separate sections, one for each level. Each section contains a description of the information that is available on that level. Also the possible actions are described.

4.4 The FILE level

This is the first (highest) level of the Type Definition Module. When the program is started as described in section 4.2, it starts on this level!

On the FILE level the user has to select on which Type Definition File(s) he wants to work. This selection is made by pressing the SELECT button. A selector will pop up with three alternatives:

PDT FILE

If this file is selected, TDM will work directly on the Pre Defined Types.

UDT FILE

Select this option to work on the local User Defined Types.

IN: BOTH/OUT: UDT

When this option is selected, TDM will read in both the PDT and the UDT file. Types from both files can be edited. However, as soon as a type is modified (or a new type inserted), it will become User Defined (and will be written to the UDT file). In this way a local version of a Pre Defined Type can be made.

After a file is selected (it's name will be shown on the Status Line), the user can start editing the types that are defined in it. To do that, the RIGHT ARROW key has to be pressed, which will activate the TYPE level.

If the LEFT ARROW key is pressed, TDM will ask the user to confirm that he wants to leave the program.

4.5 The TYPE level

On this level the Type Definition File(s) to work on have been chosen. TDM will now automatically select the first type in the current file (if there is at least one), and will display its information in the general output window. The screen will look as in figure 4.2.

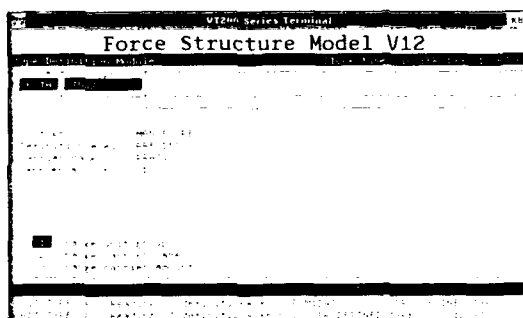


Fig. 4.2: The TYPE level

The information displayed about a type is:

Unit Group	The unit group to which this type belongs (at this moment one of the following: MANOEUVRE, ARTILLERY, C_MANOEUVRE or C_ARTILLERY).
Definition level	Whether the type is defined in the PDT or UDT file (either PRE_DEF or USER_DEF).
Carrier name	The name of the carrier (possibly empty).
Carrier amount	The number of carriers in this type.

In the bottom half of the general output window, a menu is displayed, allowing the user to choose one of the above items. The unit group, the carrier name and the carrier amount can be modified. The definition level is maintained by TDM and cannot be modified directly by the user. If BOTH files were selected on the file level, every type that is modified will get definition level USER_DEF. Use the UP and DOWN arrow keys or the numerical keys to choose an item and press return to modify it.

Using the SELECT key, a new type can be selected from the list of types defined in the file(s) that was (were) selected on the FILE level. The name of the selected type is shown on the status line and its information is displayed.

If the REMOVE key is pressed, TDM will ask the user to confirm the deletion of the currently selected type. If confirmed, the type with all its weapons/ammo and assignments will be deleted.

Press the INSERT key to add a new type. The program asks for the name of the new type and allocates space. After adding a type it will be automatically selected so the user can continue to modify its information.

The RIGHT ARROW key moves TDM to the next level. In this case TDM asks the user to which level he wants to move. Possibilities are either WEAPONS/AMMO (see sections 4.6 and 4.7) or ASSIGNMENTS (see section 4.8).

With the LEFT ARROW key returns TDM to the FILE level. TDM asks the user if he wants to save the modified types. Types will be saved in the file that was selected on the FILE level. If BOTH files were selected, only the types that are USER_DEF will be saved (in the UDT file).

4.6 The WEAPON level

On this level, a type has been selected and its weapons (and ammo) can be edited. If the current type has at least one weapon, the first weapon from the list will be selected automatically and its information will be displayed. Figure 4.3 shows what the screen looks like on this level.

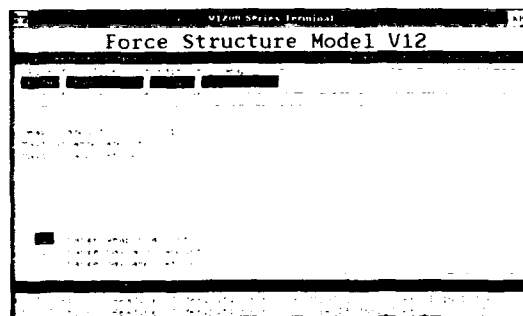


Fig. 4.3: The WEAPON level

The following information about a weapon is displayed:

Weapon amount	The number of this kind of weapon that the current type has.
Max ammo amount	The maximum space that is directly available for all ammo for this weapon.
Max ammo stock	The maximum space that is available in the stock for all ammo for this weapon.

Again, the bottom half of the output window shows a menu to select one of the above pieces of information. Use the UP and DOWN ARROW keys to select and RETURN to modify.

As with types, SELECT allows selection of a weapon from all weapons in the current type and REMOVE deletes the currently selected weapon. With INSERT a new weapon can be added. TDM offers a selection from all weapons that are defined in the Database Weapon Indicators (DBWI) for the carrier of the current type. Ofcourse also a weapon outside this DBWI list can be inserted (using the '-input-' item).

Pressing the LEFT ARROW key will make TDM return to the TYPE level, while pressing the RIGHT ARROW key activates the AMMO level (described in the next section).

4.7 The AMMO level

On this level the ammunition for the currently selected weapon can be edited. As on the other levels, the first available kind of ammunition is selected and displayed. In figure 4.4 a sample screen is shown.

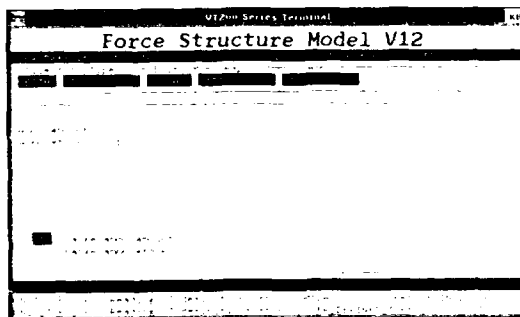


Fig. 4.4: The AMMO level

Information displayed about ammunition is:

Ammo amount	The amount of ammo that is directly available.
Ammo stock	The amount of ammo in stock.

All operations that work on the WEAPON level, work similarly on the AMMO level. UP/DOWN/RETURN select and modify information. SELECT chooses a new ammo, REMOVE deletes the currently selected kind of ammo and INSERT adds a new kind of ammo. The LEFT ARROW key returns the user to the WEAPON level. Since there's no deeper level, pressing the RIGHT ARROW key has no effect.

4.8 The ASSIGNMENTS level

On this level the list of assignments that is attached to a UNIT_TYPE can be edited. The mode of operation on this level is a bit different from the other levels.

The status line shows that the user is working on assignments and tells how many assignments are defined for the currently selected type.

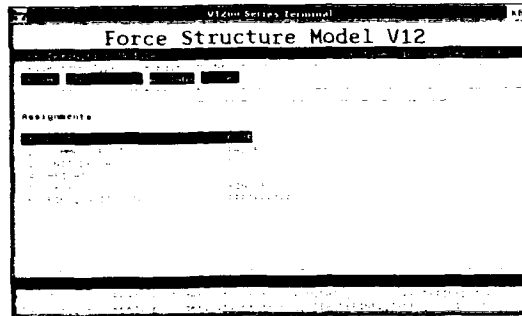


Fig. 4.5: The ASSIGNMENTS level

In the general output window a list of assignments is displayed. Figure 4.5 shows what the screen looks like on this level. This list consists of a number of lines with the following format:

num> ATTRIBUTE = VALUE

The user can go through this list using the UP and DOWN ARROW keys or the NEXT and PREV SCREEN keys. After pressing RETURN, a new value for an attribute can be given. Pressing the REMOVE key deletes an assignment (after confirmation). Pressing INSERT allows the user to add an assignment by selecting a new attribute from a list of all possible unit attributes. After selecting an attribute, its value can be entered. Currently TDM does not check whether the attribute you add belongs to the unit group of the unit.

LEFT ARROW returns TDM to the TYPE level, and since there are no deeper levels, the RIGHT ARROW key is not active.

REFERENCES

- [Bakuwel 1] An Advanced Terminal Driver (ATD) for VAX/VMS. Bakuwel, Ing. J.A., 1989. Report no. FEL-89-A352.
- [DBWI] Database Weapon Indicators. Intermediate report. PML 1986-3 (classificatie: geheim)
- [Joppe 1] Concept of the Wargame Construction Set (WCS). Joppe, Drs. W., Schagen, Drs. P.A.B. van, e.a. To be issued.
- [Schagen 1] Eindrapportage Project Force Structure Modelling. Schagen, Drs. P.A.B. van, Joppe, Drs. W. en Le Mahieu, Ir. M.J. 1989. Rapport no. FEL89A345.
- [Schagen 2] User Manual of the Force Structure Model Scenario Editor SCENE. Schagen, Drs. P.A.B. van, 1988. Report no. FEL1988-101



Drs. P.A.B. van Schagen
(group leader)



Vdg. Drs. R. Kiel
(author)

EXAMPLE PARAMETER FILES

In this appendix an example FSM_USER_FILES and FSM_USER_PARAMETERS file is given. Note that these files can contain also parameters for other programs, without affecting the performance of TDM. For clarity and brevity only TDM parameters are given.

FSM_USER_FILES

FSM_USER_PARAMETERS	FILE	[]FSM_USER_PARS
ERROR_FILE	FILE	FSM\$DAT:[000000]FSM_ERROK
EML_FILE	FILE	[]EKP\$LOG
UDT_FILE	FILE	[]USER_DEFINED_TYPES
LOG_TERM	TERMINAL	NL

[]FSM_USER_PARS

RUN_MODE	RUN_MODES	INTERACTIVE
RUN_ID	INTEGER_TYPE	101

TDM ERRORS

When working with TDM a number of things can go wrong (e.g. trying to define a type that already exists). In case of such an error, a message is printed on the bottom line of the VDU screen (the LOG line). Below a list of all possible errors is given.

- 500 Internal TDM error
- 501 No Type Definition File (UDT or PDT) selected
- 502 No Type selected
- 503 There are no Types to select from
- 504 New type-name should not be empty
- 505 No Weapon selected
- 506 There are no Weapons to select from
- 507 No Ammo selected
- 508 There's no Ammo to select from
- 509 Maximum number of weapons per carrier = \$
- 510 Current carrier already has a weapon named '\$'
- 511 Maximum number of ammos per weapon = \$
- 512 Current weapon already has ammo named '\$'
- 513 Maximum number of unit types = \$
- 514 Type '\$' is already defined
- 515 Assignment '\$' is already defined

UNCLASSIFIED
REPORT DOCUMENTATION PAGE

(MOD-NL)

1. DEFENSE REPORT NUMBER (MOD-NL) TD90-1228	2. RECIPIENT'S ACCESSION NUMBER	3. PERFORMING ORGANIZATION REPORT NUMBER FEL-90-A080
4. PROJECT TASK WORK UNIT NO. 20565	5. CONTRACT NUMBER A89KL619	6. REPORT DATE NOVEMBER 1990
7. NUMBER OF PAGES 26 (INCL. APPENDICES, EXCL. RDP + DISTRIBUTION LIST)	8. NUMBER OF REFERENCES 5	9. TYPE OF REPORT AND DATES COVERED PROGRAM MANUAL

10. TITLE AND SUBTITLE
USER MANUAL OF THE FORCE STRUCTURE MODEL UNIT TYPE EDITOR TDM

11. AUTHOR(S)
VDG DRS R. KIEL

12. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TNO PHYSICS AND ELECTRONICS LABORATORY, P.O. BOX 96864, 2509 JG THE HAGUE
OUDE WAALSDORPERWEG 63, THE HAGUE, THE NETHERLANDS

13. SPONSORING/MONITORING AGENCY NAME(S)
TNO DIVISION OF NATIONAL DEFENSE RESEARCH, THE NETHERLANDS

14. SUPPLEMENTARY NOTES
THE PHYSICS AND ELECTRONICS LABORATORY IS PART OF THE NETHERLANDS ORGANIZATION FOR
APPLIED SCIENTIFIC RESEARCH

15. ABSTRACT (MAXIMUM 200 WORDS, 1044 POSITIONS)

THE TYPE DEFINITION MODULE TDM IS PART OF THE COMBAT SIMULATION MODEL FORCE STRUCTURE
MODEL FSM AND HAS BEEN DEVELOPED UNDER CONTRACT A89KL619. THE PURPOSE OF TDM IS TO
PROVIDE A WAY OF INTERACTIVELY CREATING AND MODIFYING SO-CALLED UNIT_TYPES. THESE
UNIT_TYPES ARE USED AS TEMPLATES FOR CREATING THE UNITS FOR A FSM SIMULATION RUN. THE
UNIT_TYPES ARE STORED IN TWO TYPE DEFINITION FILES: THE SYSTEMWIDE AVAILABLE PRE_DEFINED_TYPES
AND THE LOCALLY AVAILABLE USER_DEFINED_TYPES. THE REPORT SERVES AS THE USER MANUAL FOR TDM.

16. DESCRIPTORS
COMPUTER PROGRAM
WARGAME

IDENTIFIERS

17a. SECURITY CLASSIFICATION
(OF REPORT)
UNCLASSIFIED

17b. SECURITY CLASSIFICATION
(OF PAGE)
UNCLASSIFIED

17c. SECURITY CLASSIFICATION
(OF ABSTRACT)
UNCLASSIFIED

18. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

17d. SECURITY CLASSIFICATION
(OF TITLES)
UNCLASSIFIED

UNCLASSIFIED